Infrared Remote Control Experiment

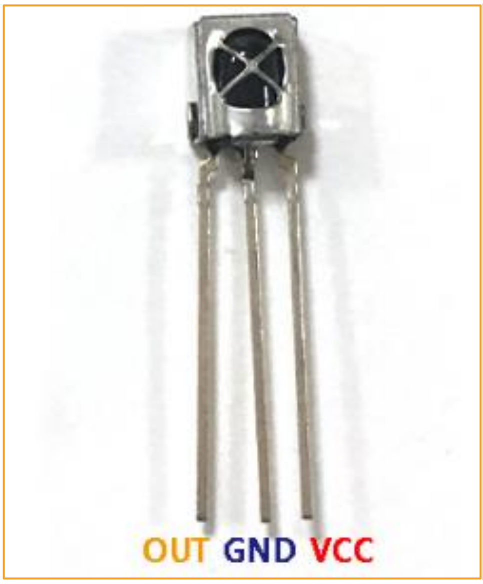
Introduce of Infrared Remote Control

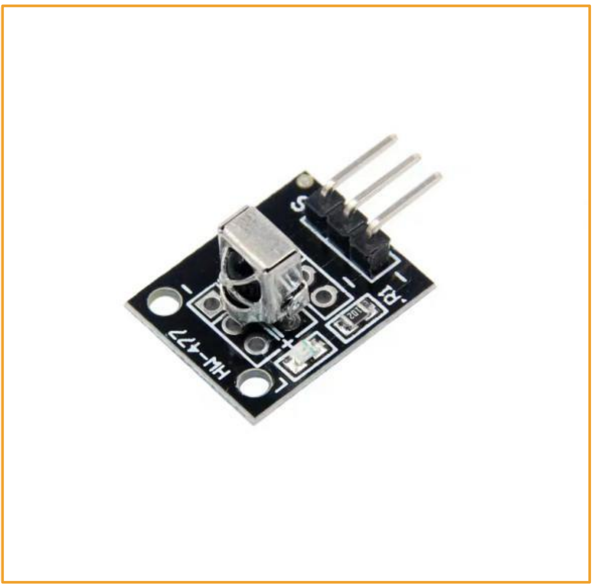
The signal emitted by the infrared remote control is a series of binary pulses. In order to protect it from interference of other infrared signals during wireless transmission, it is usually modulated on a specific carrier frequency - 38kHZ and then transmitted through the infrared emitting diode. The integrated reception and modulation of infrared sensors are infrared. An infrared receiver filters out the clutter, picks up the signal at that particular frequency and converts it back into a binary pulse code, or demodulation.

Working Principle

The built-in receiving tube converts the light signal emitted by infrared transmitting tube to weak electrical signal, the signal is amplified through the internal IC, then restored to original code emitted by the infrared remote control through the automatic gain control, band-pass filtering, demodulation, waveform shaping, and input to the decoding circuit on electric appliance through the output pin of the receiving header

Infrared receiver header has three pins: connecting VOUT to analog interface, GND to the GND onboard and VCC to + 5 v.





Experiment Purpose

The goal is to encode all the keys of the remote control via Ardunio and display them on the serial port.

Experiment Principle

The encoding of a remote control is required to be learned first if we want to decode it. The control code used in this product is: NEC protocol.

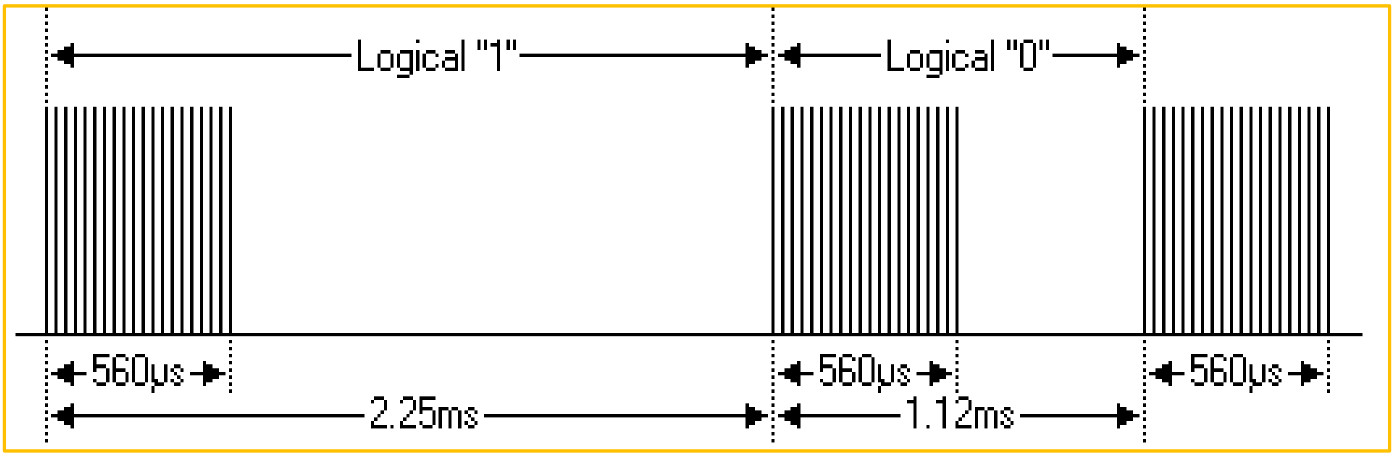
The following will introduce the NEC protocol:

Introduction of NEC

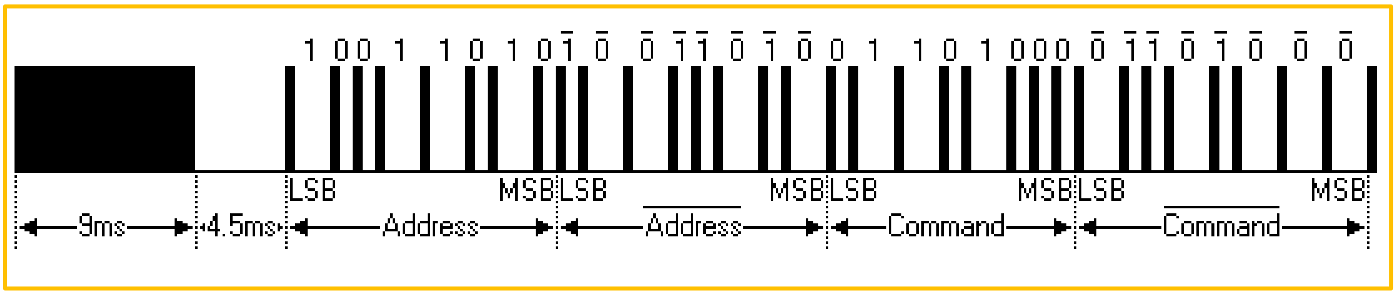
Characteristics

- 8 address bits, 8 command bits
- Address bits and command bits are transmitted twice in order to ensure reliability
- Pulse-position modulation
- Carrier frequency 38kHz
- Every bit lasts 1.125ms or 2.25ms

The definitions of logic 0 and 1 are as below

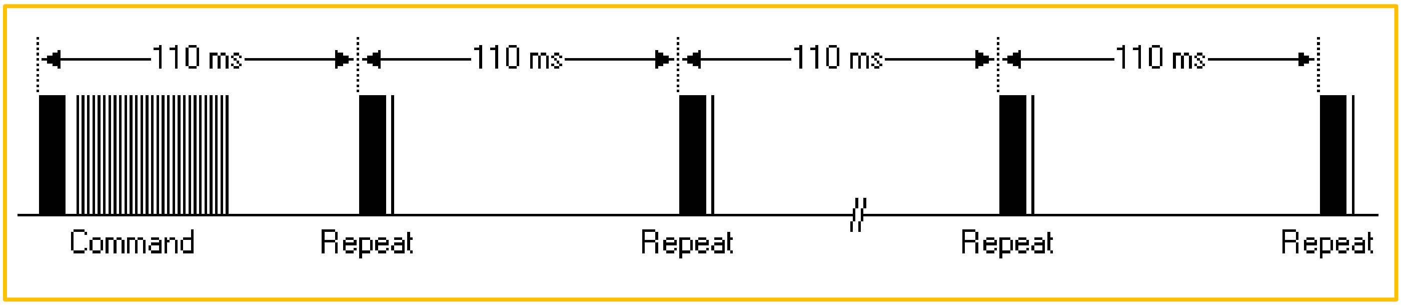


Transmitted pulse which the pressed button released immediately


1 0 0 1 1 0 1 0 $\bar{1}$ $\bar{0}$ $\bar{0}$ $\bar{1}$ $\bar{1}$ $\bar{0}$ $\bar{1}$ $\bar{0}$ 0 1 1 0 1 0 0 0 $\bar{0}$ $\bar{1}$ $\bar{1}$ $\bar{0}$ $\bar{1}$ $\bar{0}$ $\bar{0}$ $\bar{0}$
LSB
MSB
LSB
MSB
LSB
MSB
LSB
MSB
9ms
4.5ms
Address
$\overline{\text{Address}}$
Command
$\overline{\text{Command}}$

The messages start at the 9ms high level, then the 4.5ms low level, along with the address code and command code. The address and command are transmitted twice. All bits are flipped in the second transmission, which can be used to confirm the received message. The total transfer time is constant because each bit reverses the length repeatedly.

Transmitted pulse which the pressed button last for a while



Even a button on the remote control is pressed again, the command is transmitted only once. When the button has been pressing, the first 110ms pulse is same as above, then the same code is transmitted every 110ms. The next repeated code consists of a 9ms high level pulse, a 2.25ms low level pulse and a 560μs high level pulse.

Notice: When the pulse received by the integrative header, the header needs to decode,amplify and shape the signal. So we should notice that the output is high level when the infrared signal is absent, otherwise, the output is low level, so the output signal level is reversed in transmitter. We can see the pulse of receiver through the oscilloscope and understand the program by waveform.

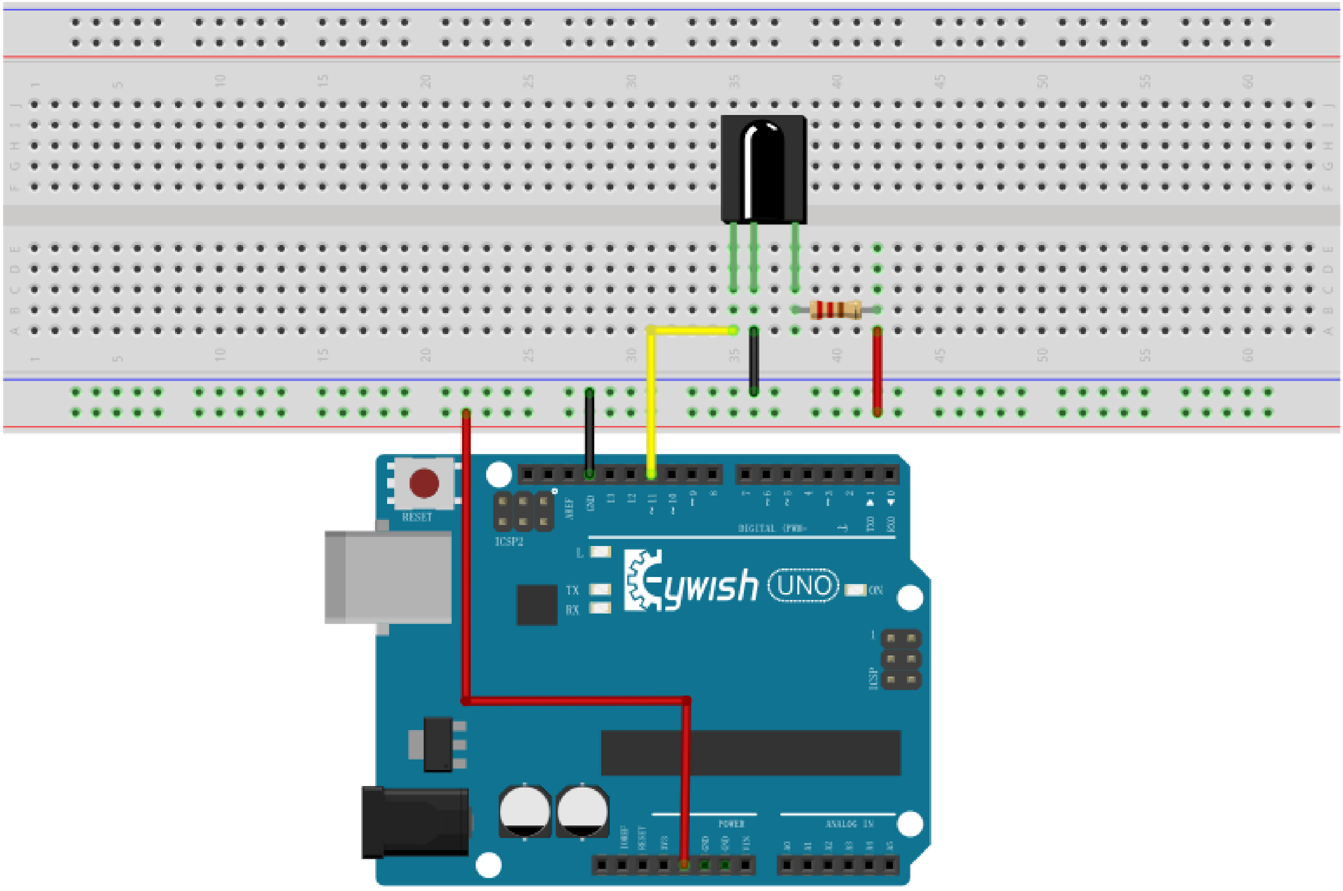
Component List

- Keywish Arduino UNO R3 mainboard
- Breadboard
- USB cable
- NEC infrared remote control * 1
- The integrative infrared receiving header *1
- 1k Resistor *1
- Several jumper wires

Wiring of Circuit

Infrared receiver module connection:

Infrared receiver module	Arduino
G	GND
R	5V
Y	12

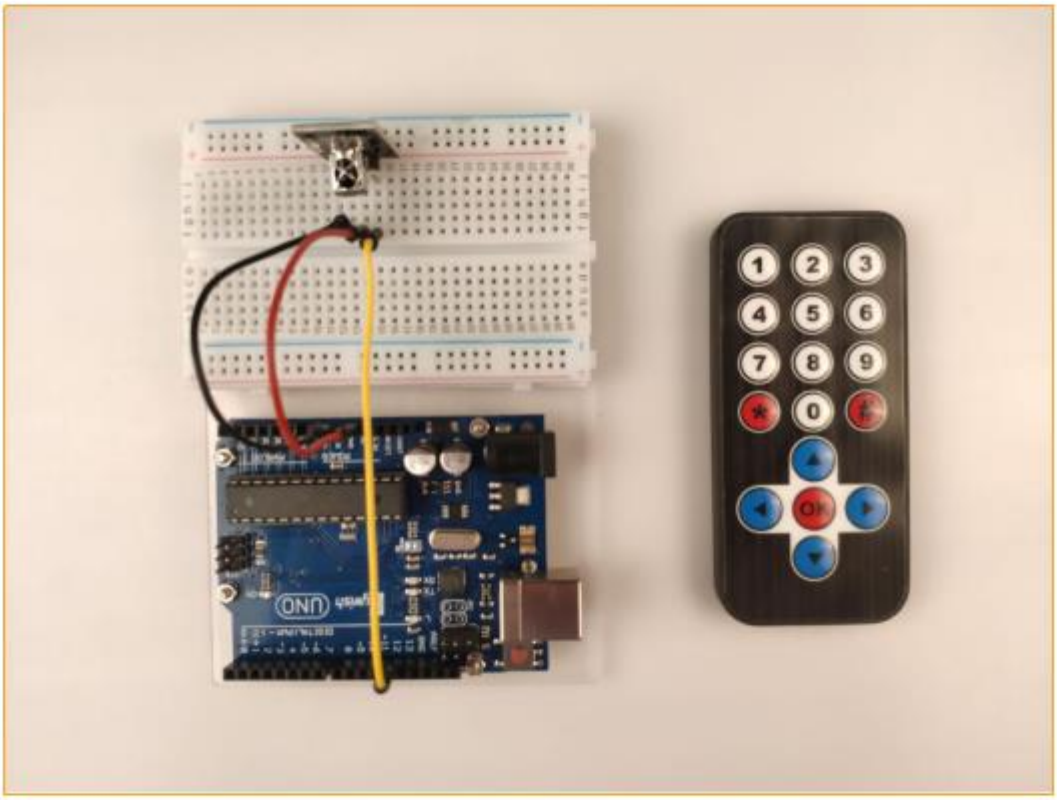

RESET
ICSP2
AREF
GND
DIGITAL (PWM~)
TX
RX
L
Keywish
UNO
ON
ICSP
IOREF
RESET
POWER
ANALOG IN

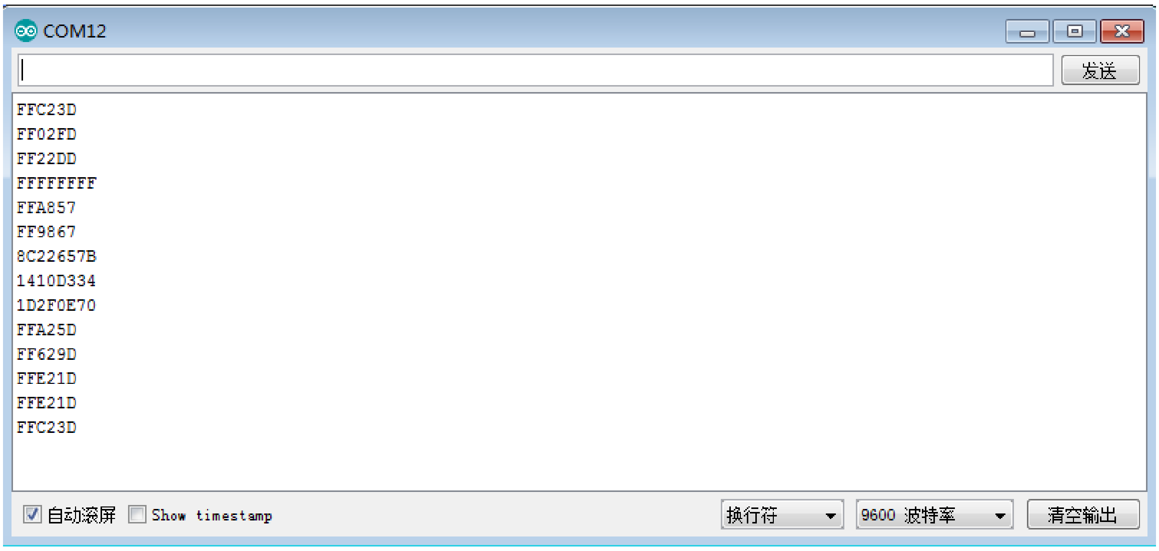
Code

```
#include "IRremote.h"
int RECV_PIN = 12;
IRrecv irrecv(RECV_PIN);
decode_results results;
void setup()
{
    Serial.begin(9600);
    irrecv.enableIRIn();
}

void loop() {
    if (irrecv.decode(&results)) {
        Serial.println(results.value, HEX);
        irrecv.resume();
    }
    delay(120);
}
```

Experiment Result





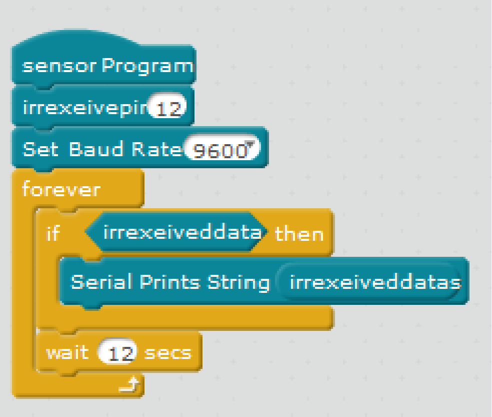
```
FFC23D
FF02FD
FF22DD
FFFFFFFF
FFA857
FF9867
8C22657B
1410D334
1D2F0E70
FFA25D
FF629D
FFE21D
FFE21D
FFC23D
```

☑ 自动滚屏 □ Show timestamp 换行符 9600 波特率

Ardunio encodes all the keys of the remote control. Pressing the button you want to be on the infrared remote control will show the button you pressed on the serial port monitor. The distance of the infrared remote control is required not to be far away, followed by aiming at the receiver head.

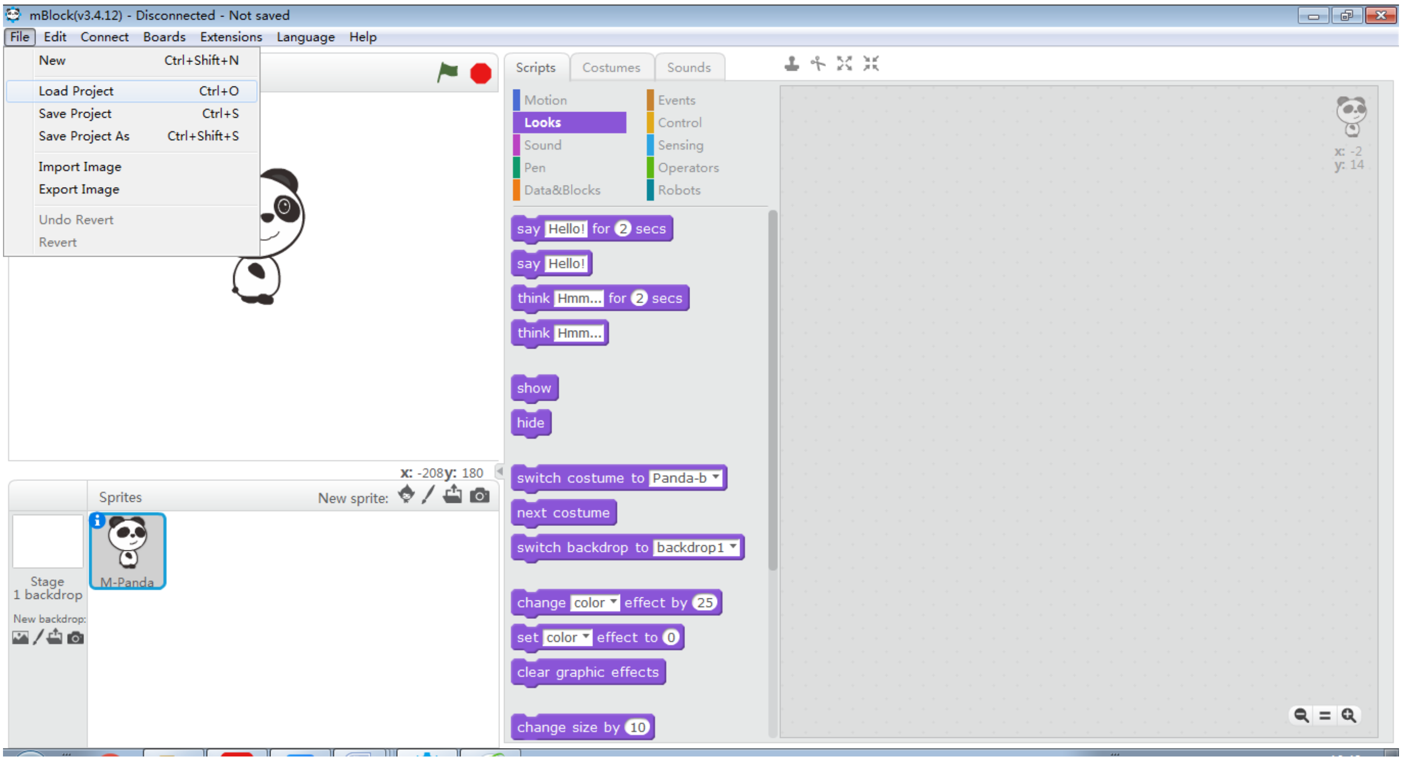
MBlock graphical programming program

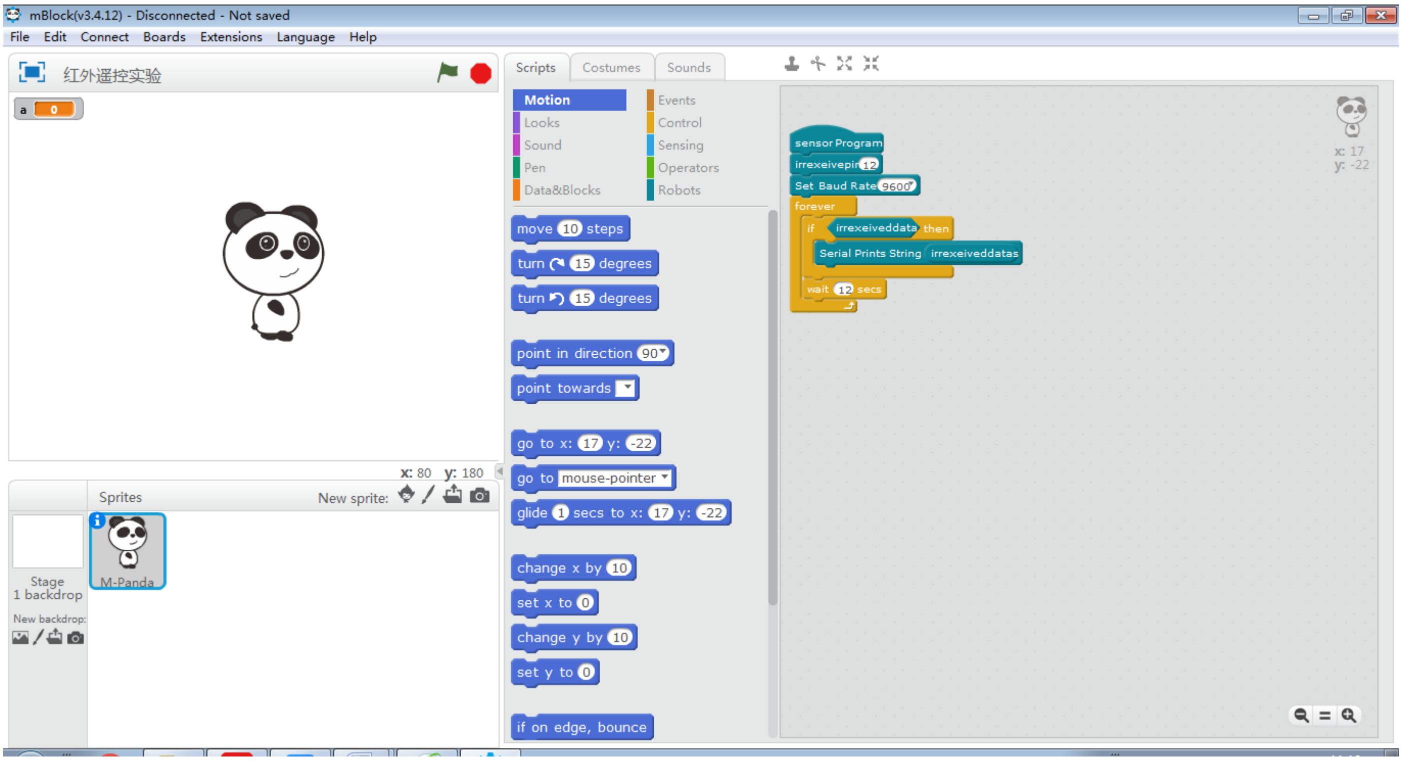
The program written by mBlock is shown in the figure below:



```
sensor Program
irrexeivepin 12
Set Baud Rate 9600
forever
    if irrexeiveddata then
        Serial Prints String irrexeiveddatas
    wait 12 secs
```

You can also open the program file directly with mblock, which is a. Sb2 file. Here are the steps to open it:

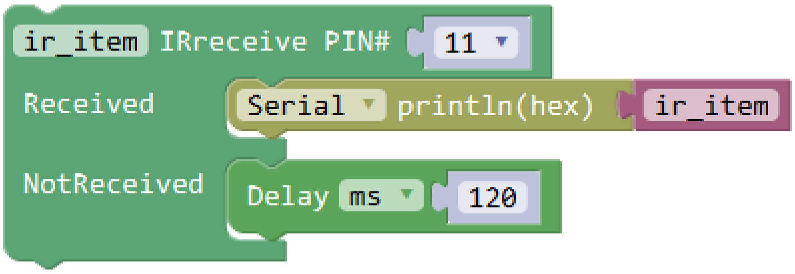




```
sensor Program
irrexeivepin 12
Set Baud Rate 9600
forever
    if irrexeiveddata then
        Serial Prints String irrexeiveddatas
    wait 12 secs
```

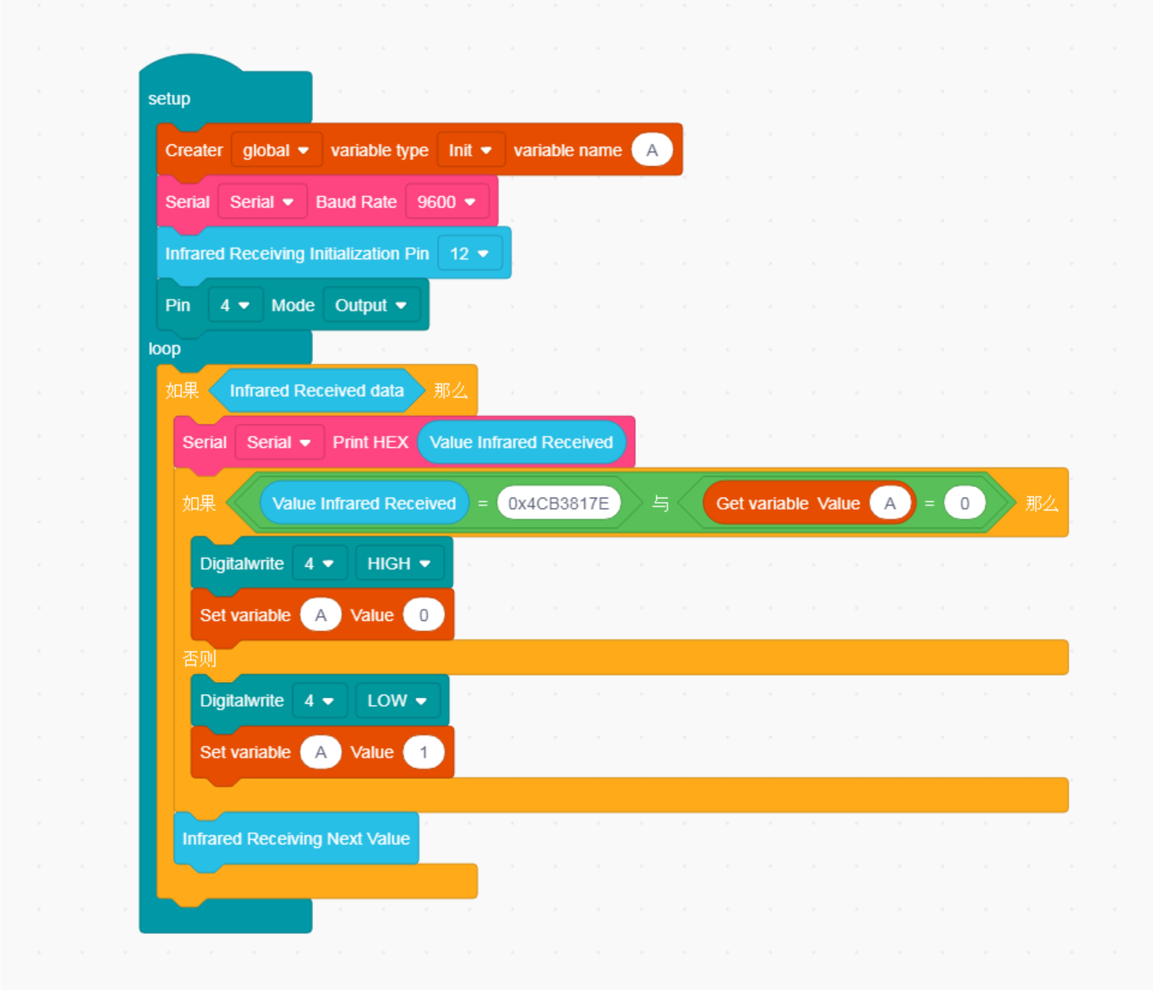
Mixly graphical programming

Mixly programmed the infrared remote control program as shown in the figure below:



```
ir_item IRreceive PIN# 11
Received     Serial println(hex) ir_item
NotReceived  Delay ms 120
```

MagicBlock graphical programming



```
setup
  Creater global variable type Init variable name A
  Serial Serial Baud Rate 9600
  Infrared Receiving Initialization Pin 12
  Pin 4 Mode Output
loop
  如果 Infrared Received data 那么
    Serial Serial Print HEX Value Infrared Received
    如果 Value Infrared Received = 0x4CB3817E 与 Get variable Value A = 0 那么
      Digitalwrite 4 HIGH
      Set variable A Value 0
    否则
      Digitalwrite 4 LOW
      Set variable A Value 1
    Infrared Receiving Next Value
```